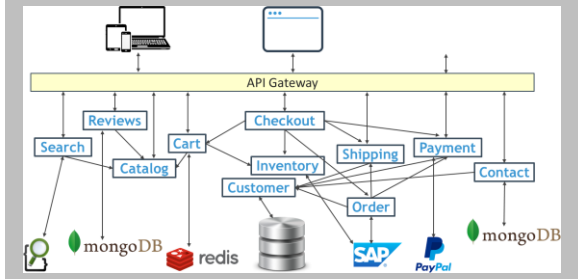
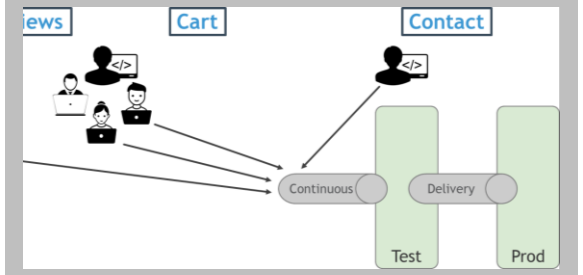
Recapitulando

"Uma aplicação única, desenvolvida como um conjunto de pequenos serviços, cada um executando o seu próprio processo e se comunicando com mecanismos leves, geralmente por API's REST através do protocolo HTTP".

– Martin Fowler



O que são Microsserviços



As Características chave dos Microsserviços

Monólitos x Microsserviços

Os Prós e Contras dos Microsserviços

O que aprenderemos ...

- Introdução ao estilo arquitetural de microsserviços
  - Impacto
- O que são microsserviços?
- As características chave dos microsserviços
- Benefícios dos microsserviços
- Diferenças entre as arquiteturas monolítica e de microsserviços
- Prós e contras

O que são Microsserviços?

- Microsserviços são uma Verdadeira "Febre";
- Os microsserviços são melhor definidos como:
  - Um estilo arquitetural;
  - Uma alternativa às aplicações "monolíticas" tradicionais;
  - As aplicações são implementadas como um conjunto de pequenos serviços, cada um dos quais sendo executado como um processo independente e cada um dos quais se comunicando com os outros por meio de algum tipo de protocolo bem conhecido;
    - Microsserviços trazem vários benefícios e riscos que devem ser considerados

Definições dos especialistas

- *"Uma aplicação única, desenvolvida como um conjunto de pequenos serviços, cada um executando o seu próprio processo e se comunicando com mecanismos leves, geralmente por API's REST através do protocolo HTTP".*

  – Martin Fowler

- *"Microsserviços são o bom e velho SOA otimizado"*.

  – Adrian Cockcroft – Netflix

API – Application Programming Interface
SOA – Service Oriented Architecture

Microsserviços – A Nossa Definição

- São uma aplicação única, desenvolvida como um conjunto de pequenos serviços
  - (em vez de um único aplicativo monolítico)
- ... cada um executando o seu próprio processo
  - (não apenas módulos / componentes dentro de um único executável)
- ... e se comunicando com mecanismos leves
  - (como HTTP e REST, ou mensagens AMQP etc)
- Escrito, implantado, dimensionado e mantido separadamente
  - (potencialmente em diferentes linguagens)
- São usados para encapsular funcionalidades ou regras de negócios
  - (em oposição ao encapsulamento natural da linguagem como pacotes, classes, jars etc)
- Podem ser substituídos e atualizados de forma independente

O que Microsserviços não são:

- O mesmo que SOA;
  - SOA diz respeito à integração de várias aplicações corporativas. Já os microsserviços são a decomposição de uma única aplicação em múltiplas aplicações menores trabalhando em conjunto.
- Uma bala de prata;
  - Um estilo arquitetural.
- Novidade! Você pode estar usando microsserviços agora e não sabe!

O que Microsserviços não são:

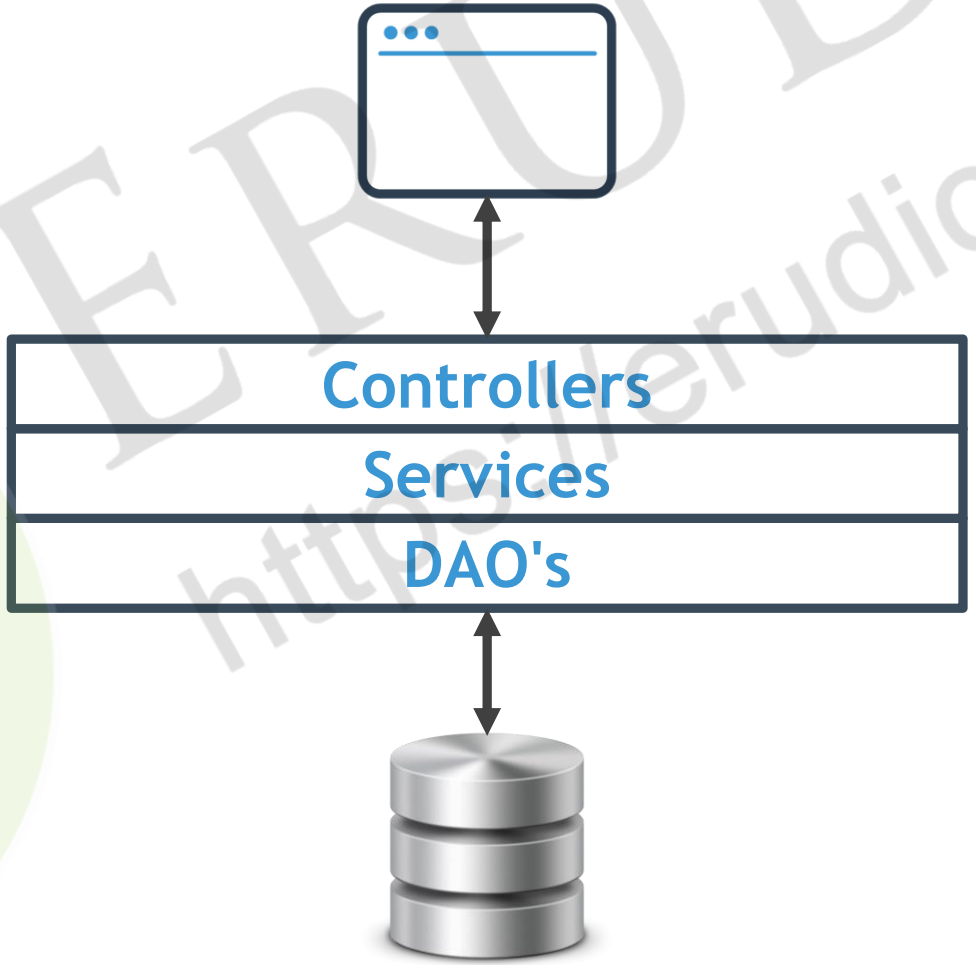
- Twitter migrou de monólitos Ruby/Rails para Microsserviços;
- Facebook migrou de monólitos em PHP para microsserviços;
- Netflix migrou de monólitos Java para microsserviços.

Exemplo de Microsserviços

- Imagine uma aplicação de carrinho de compras;
  - Métodos para:
    - Pesquisar produtos ;
    - Catálogo de produtos;
    - Gerenciamento de estoque;
    - Carrinhos de compras;
    - Checkout;
    - etc.
- E como fica com microsserviços?

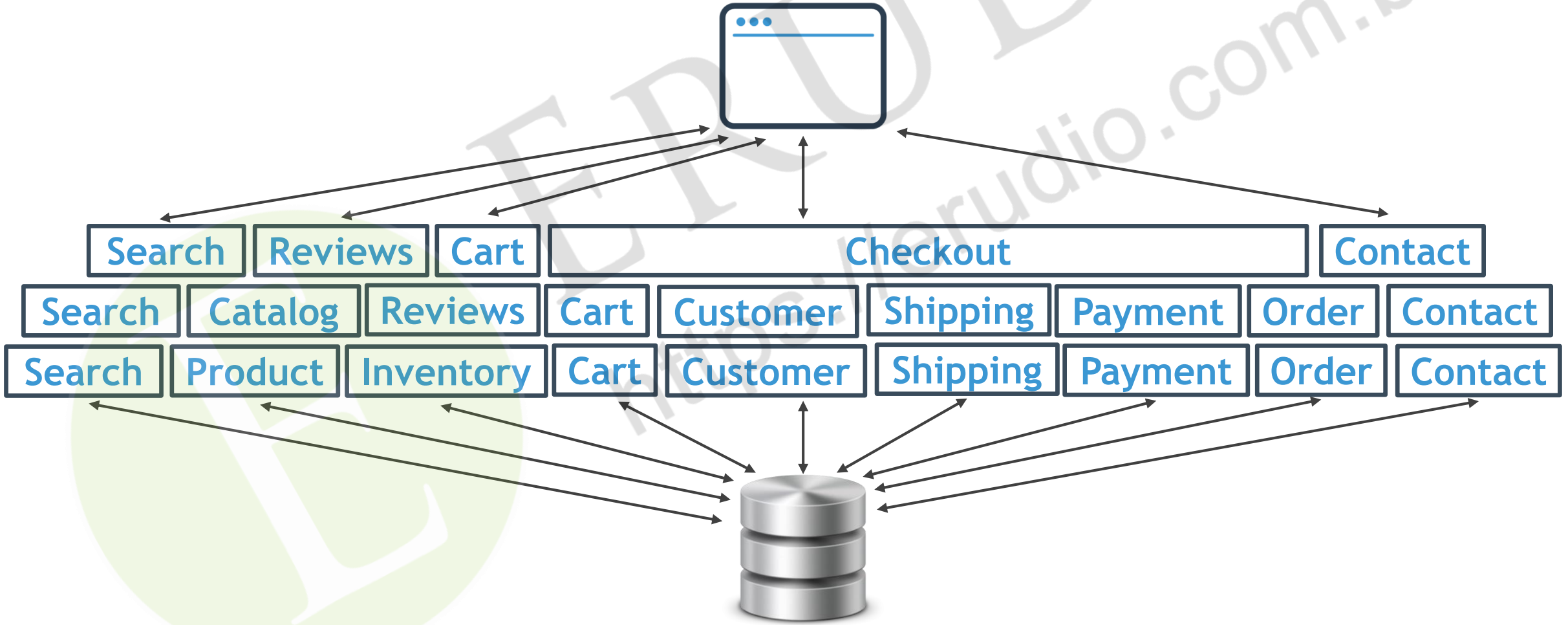
Exemplo de uma aplicação monolítica

- Aplicação de carrinho de compras monolítico:



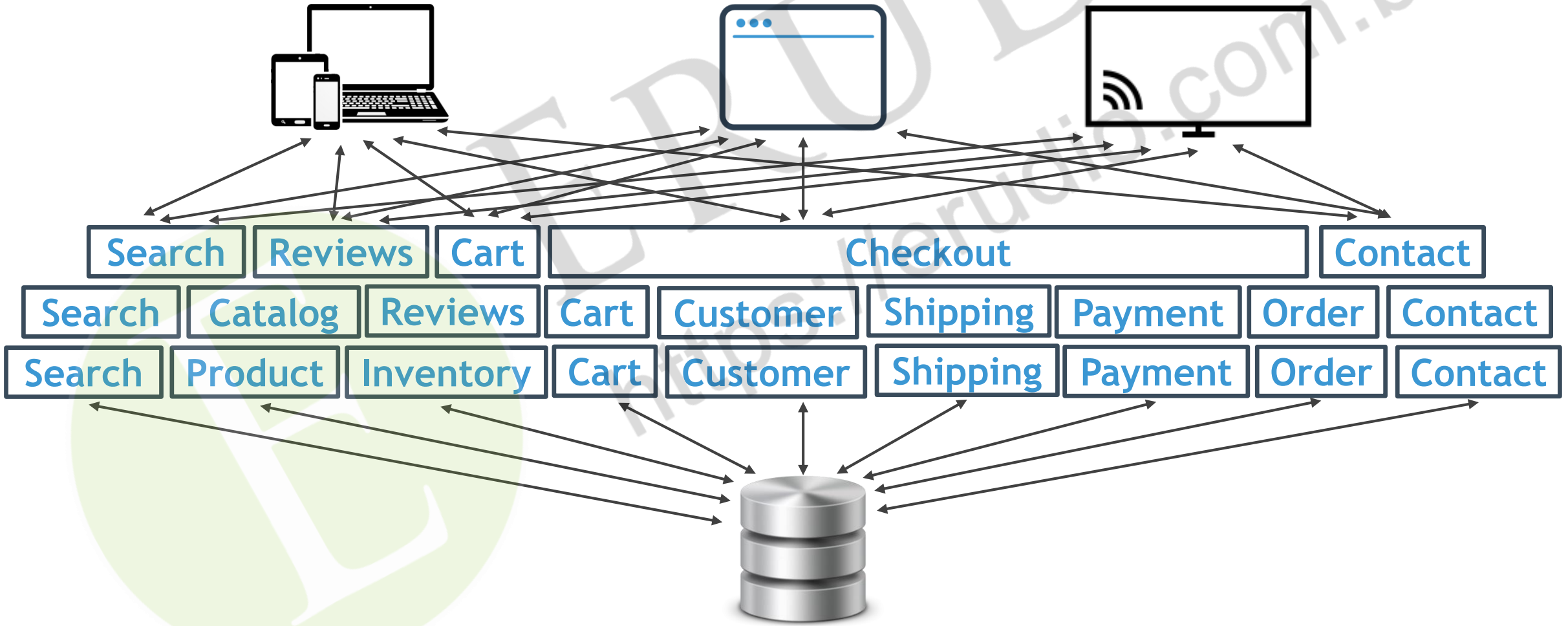
Exemplo de uma aplicação monolítica

- Entendendo a Arquitetura Monolítica



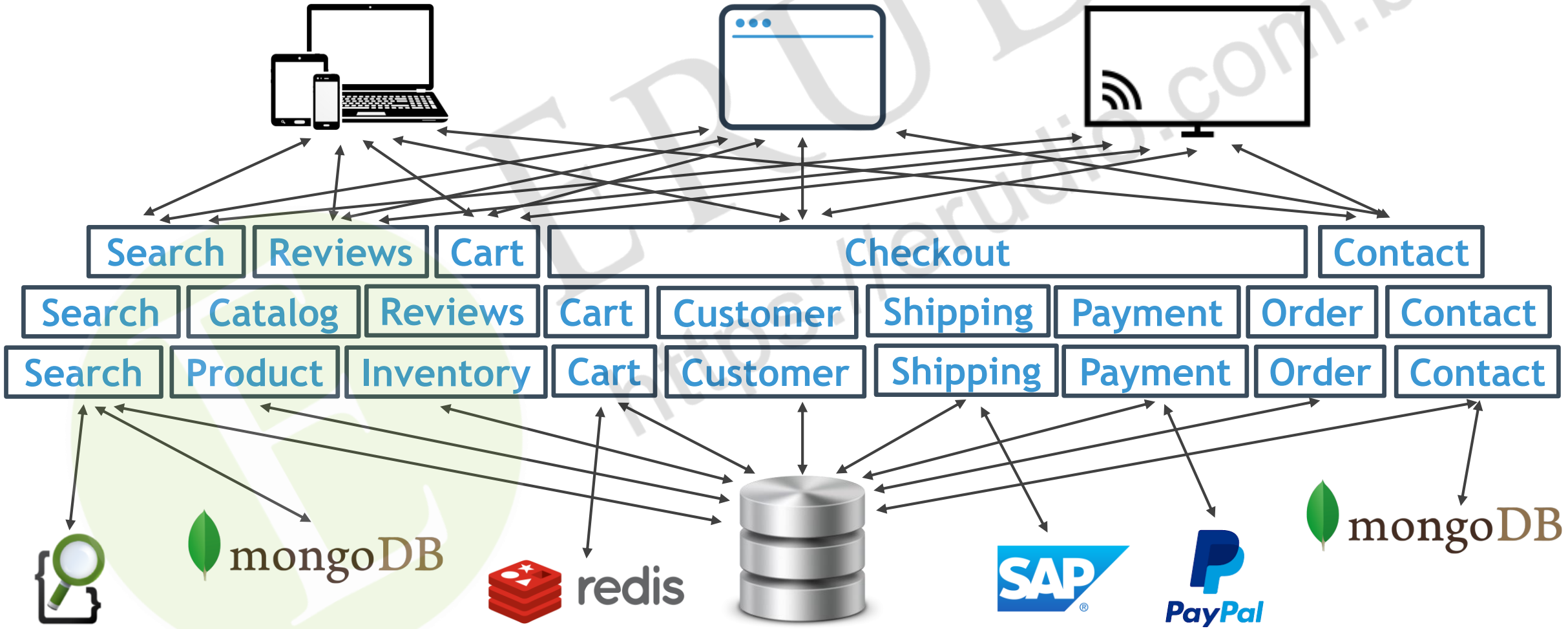
Os Desafios das Aplicações Monolíticas

- Novos tipos de aplicativos cliente



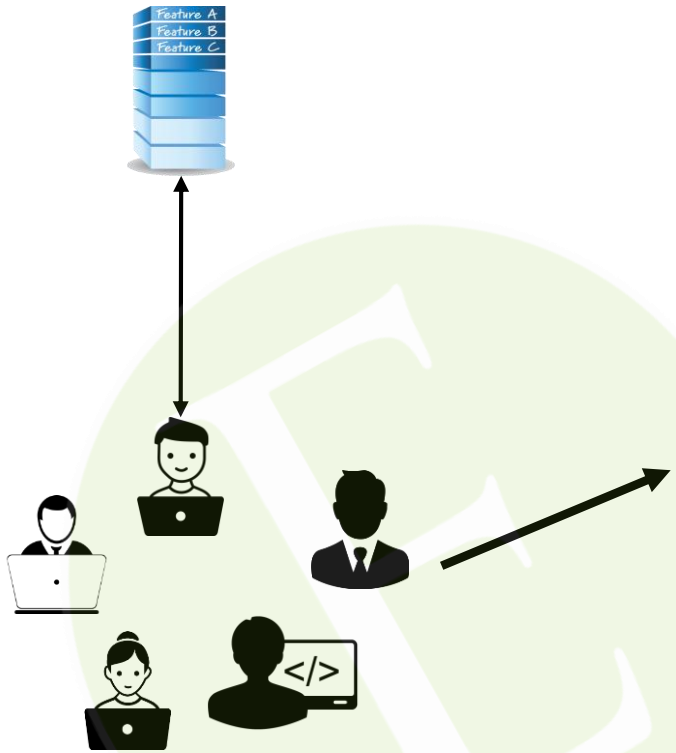
Os Desafios das Aplicações Monolíticas

- Novos tipos de persistência / serviços

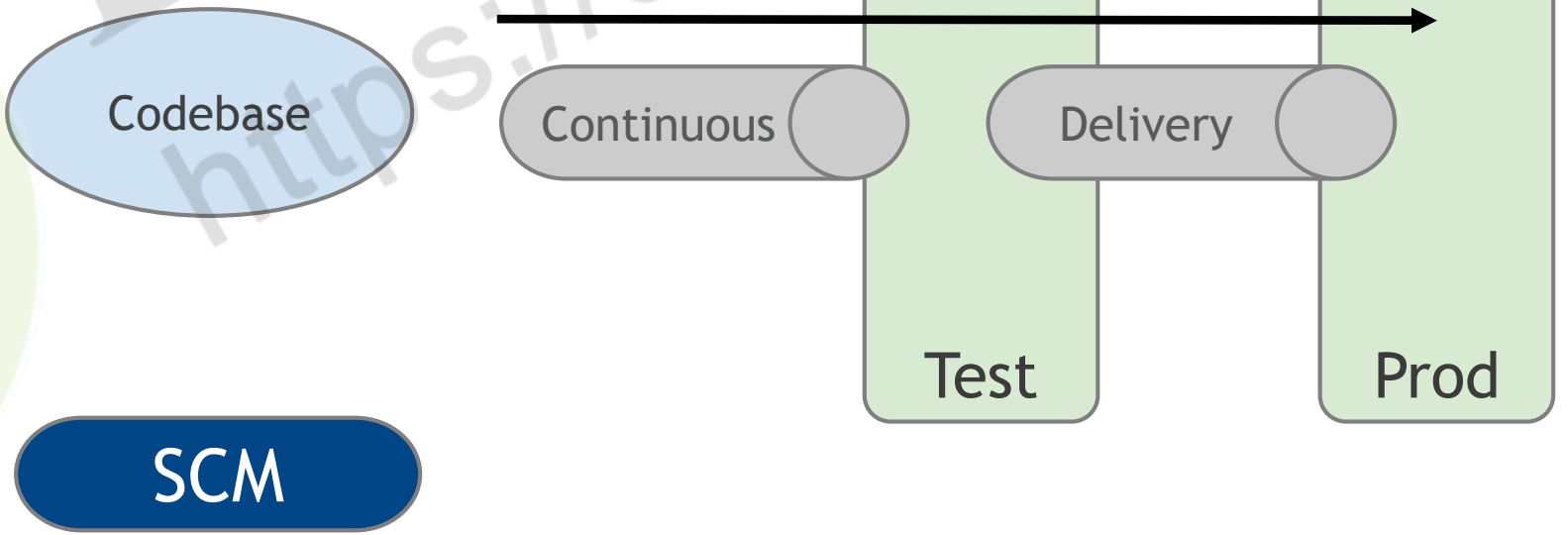


Os Desafios das Aplicações Monolíticas

- Base de código única, implantação, controle de versão, tamanho da equipe

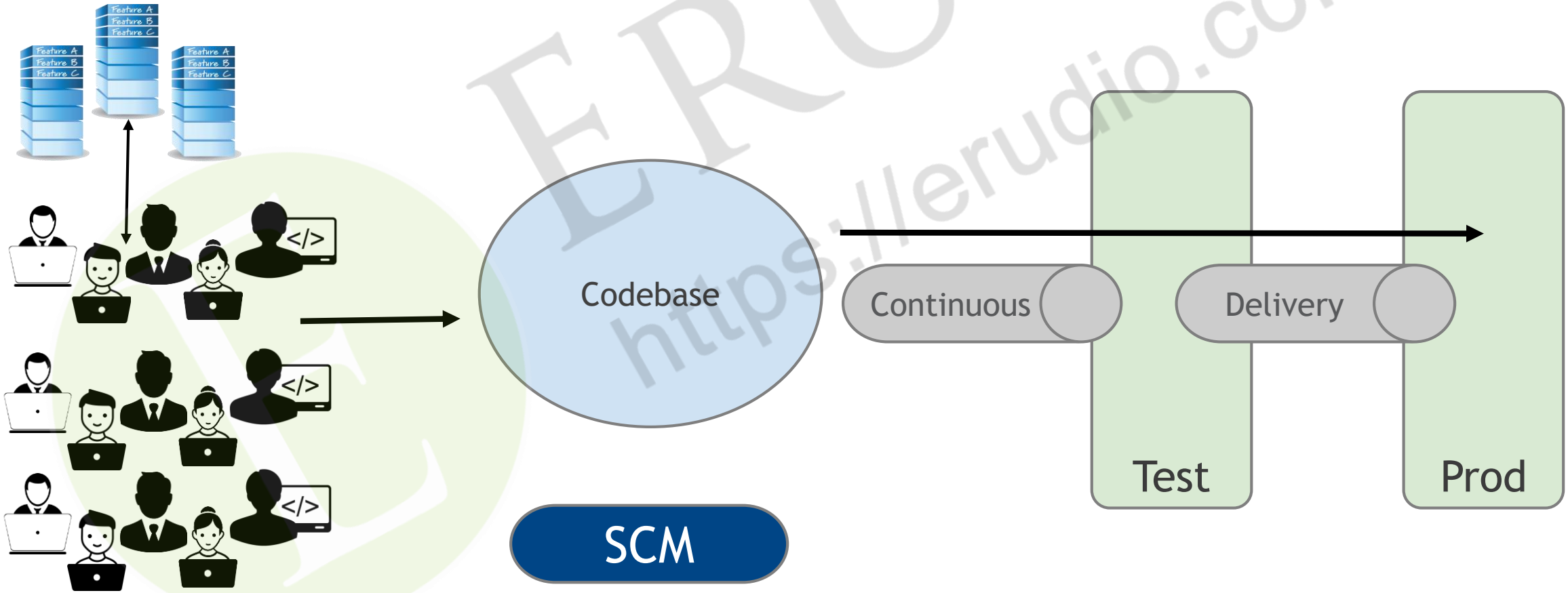


SCM – Source Code Management



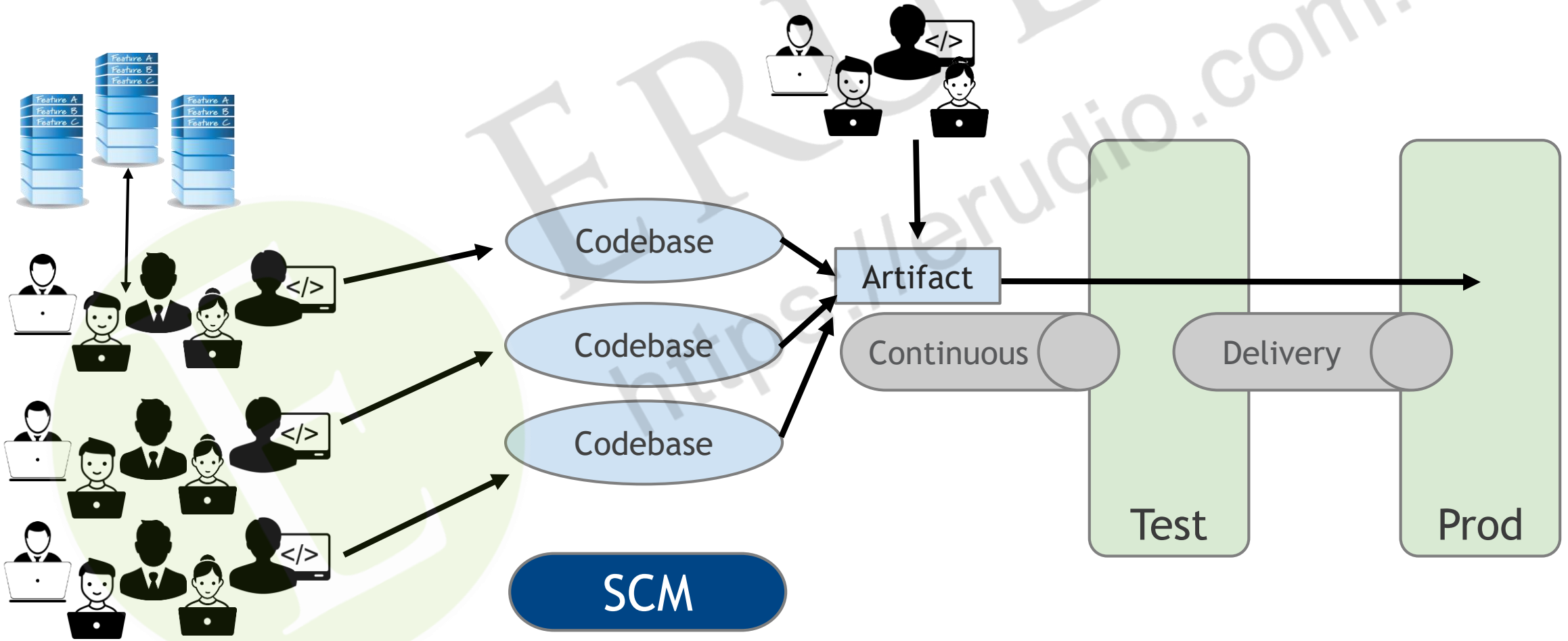
Os Desafios das Aplicações Monolíticas

- Base de código única, implantação, controle de versão, tamanho da equipe



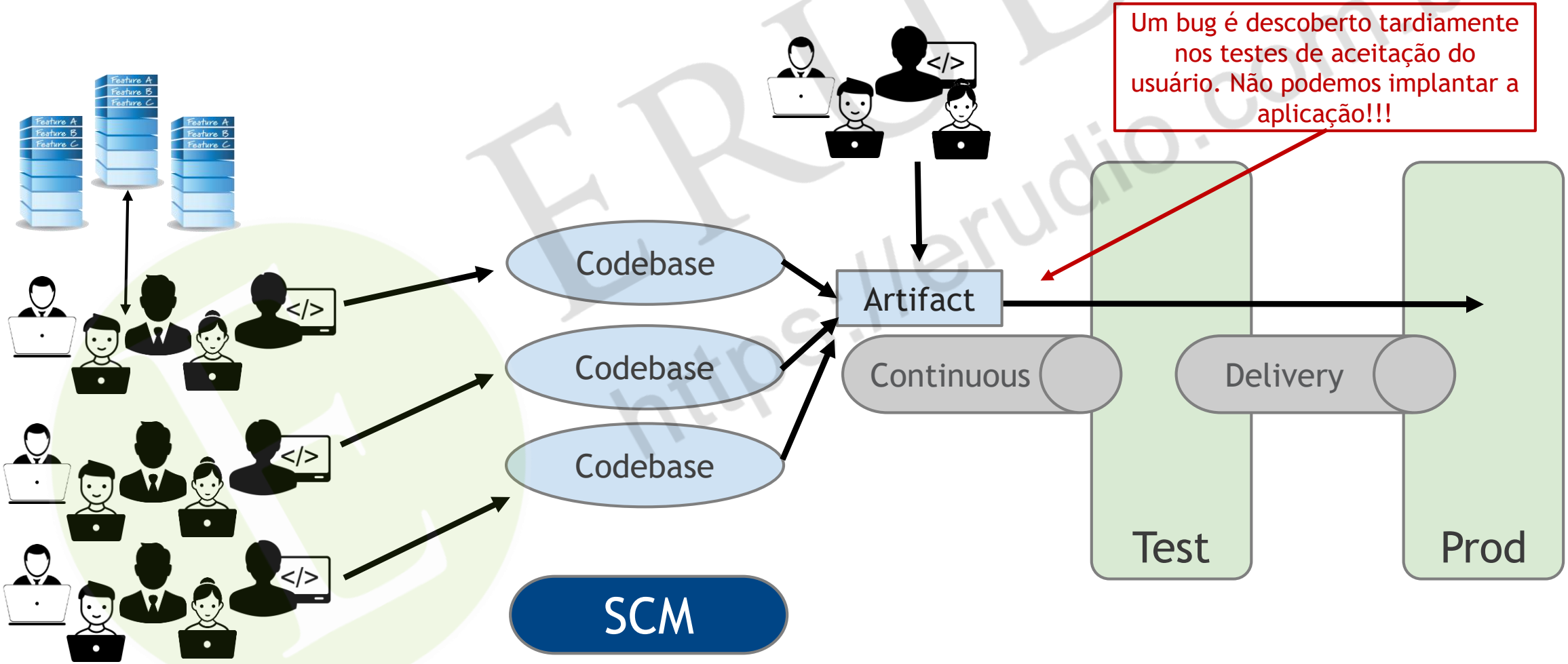
Os Desafios das Aplicações Monolíticas

- Times menores e um time de merge para "juntar as peças"



Os Desafios das Aplicações Monolíticas

- Times menores e um time de merge para "juntar as peças"



Entendendo as Aplicações Monolíticas

- Um único executável da aplicação;
  - Fácil de compreender mas, difícil de assimilar;
  - Deve ser escrito em uma única linguagem.
- Modularização de acordo com linguagem ou framework ;
  - Limitada às estruturas da linguagem (pacotes, classes, jars, wars, functions, namespaces, frameworks etc);
  - Diferentes serviços / tecnologias de armazenamento;
    - RDBMS, mensageria, e-mail etc.

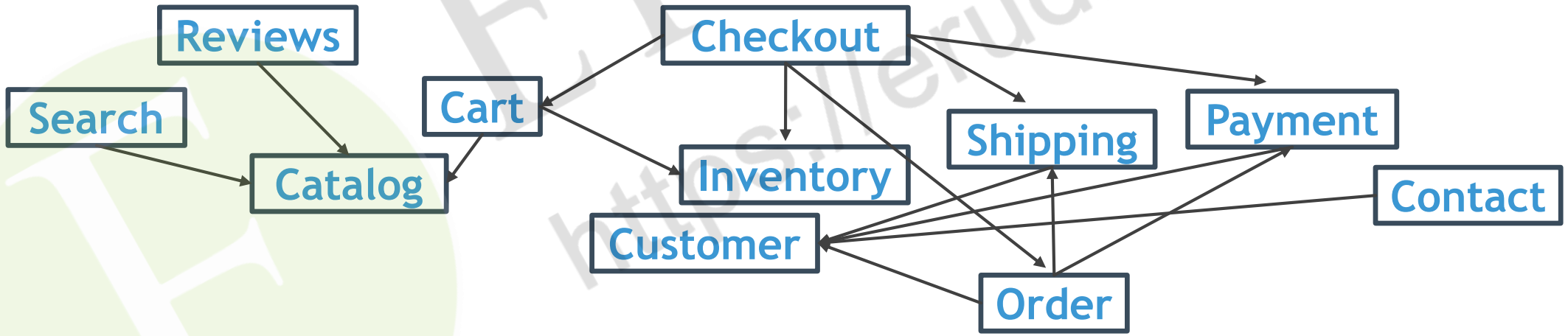
Vantagens das Aplicações Monolíticas

- Fácil de compreender (mas não de assimilar)
- Fácil de testar como uma única unidade (até certo ponto)
- Fácil de implantar como uma única unidade.
- Fácil de gerenciar (até certo ponto)
- Fácil de gerenciar mudanças (até certo ponto)
- Fácil de escalar (se tomarmos os devidos cuidados)
- Complexidade gerenciada de acordo com as estruturas da linguagem.

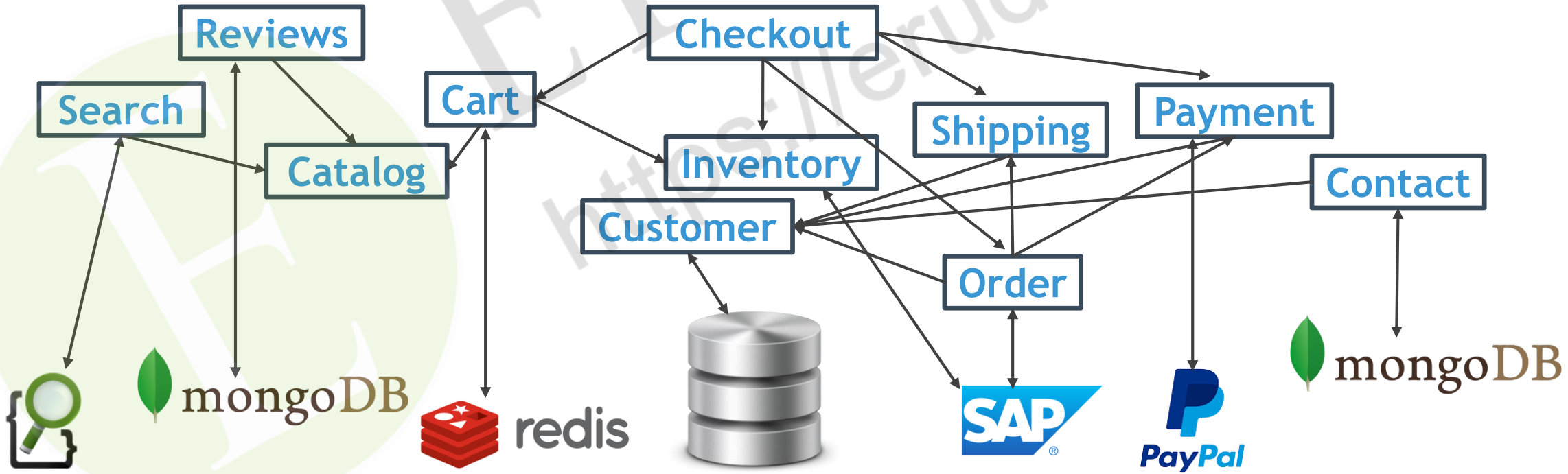
Desvantagens das Aplicações Monolíticas

- Estarmos presos a uma linguagem;
  - Nossa aplicação está presa à uma única stack de tecnologia. Não temos liberdade para experimentar tecnologias diferentes.
- Assimilação
  - Um único desenvolvedor não consegue assimilar uma grande base de código
  - Uma única equipe não pode gerenciar uma única aplicação grande
    - "Regra das 2 Pizzas" da Amazon
- Implantação como unidade única
  - Não é possível implantar independentemente uma única mudança em um único componente.
  - As mudanças são “reféns” de outras mudanças

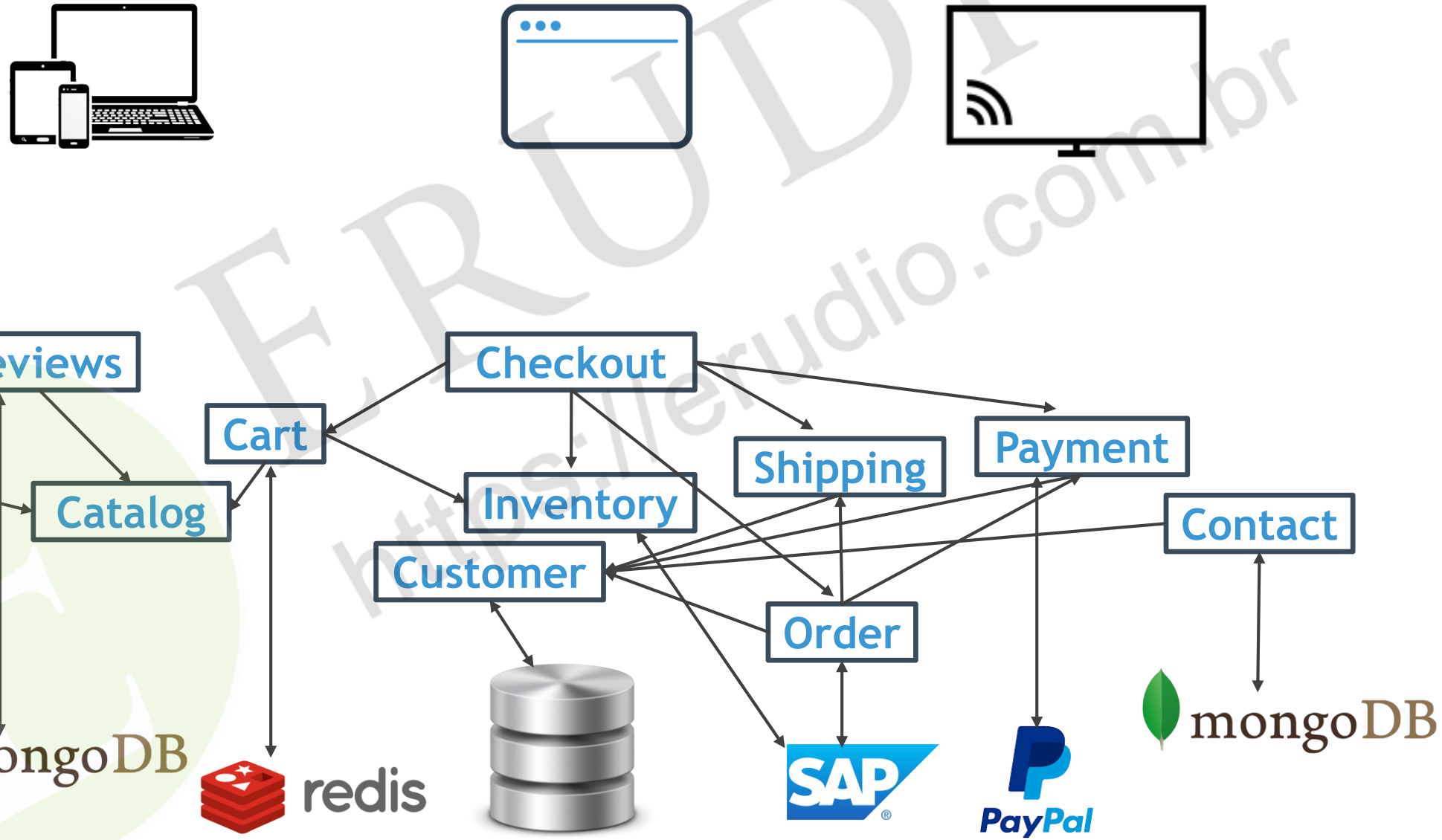
Entendendo uma Arquitetura de Microsserviços



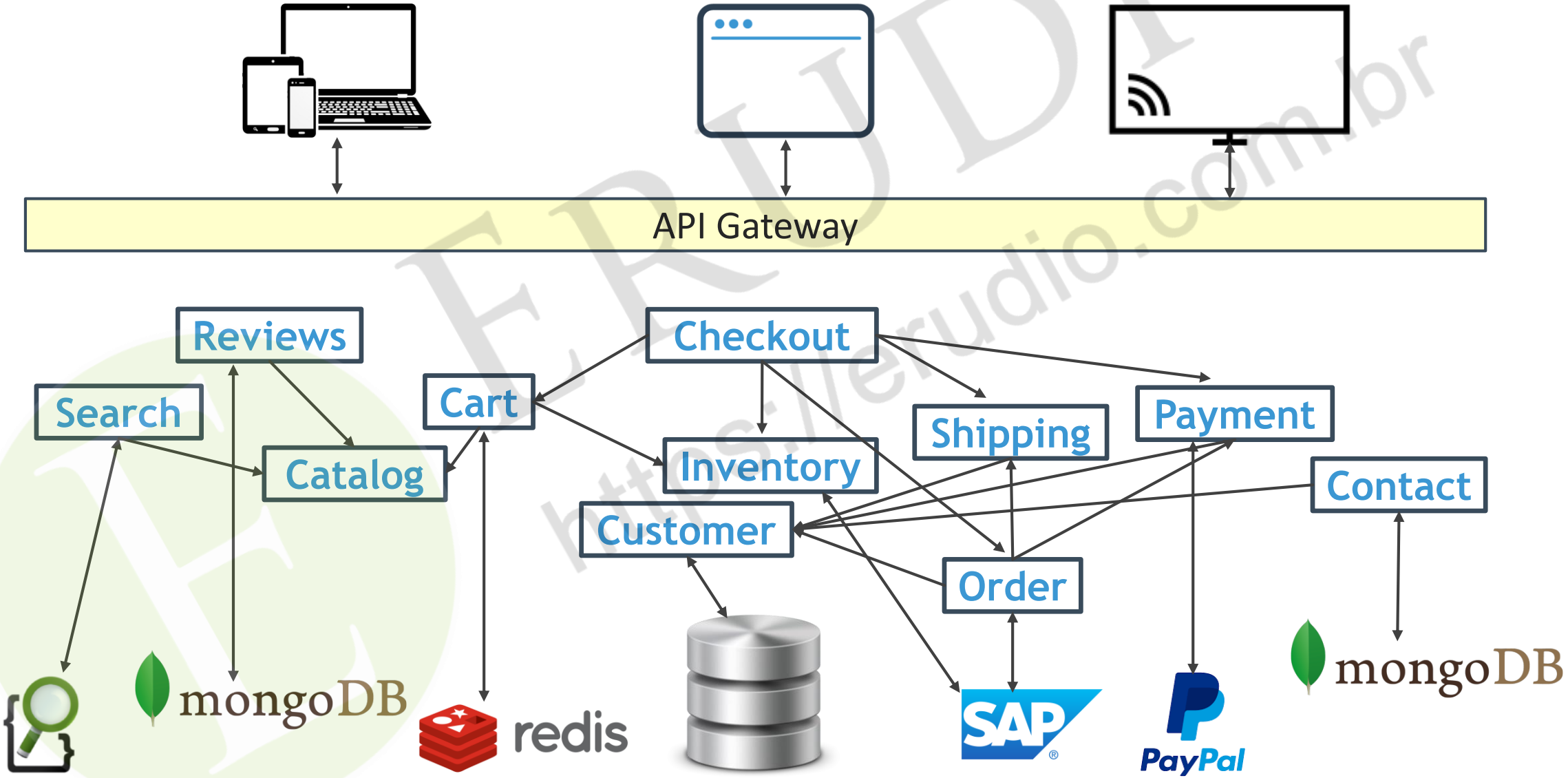
Entendendo uma Arquitetura de Microsserviços



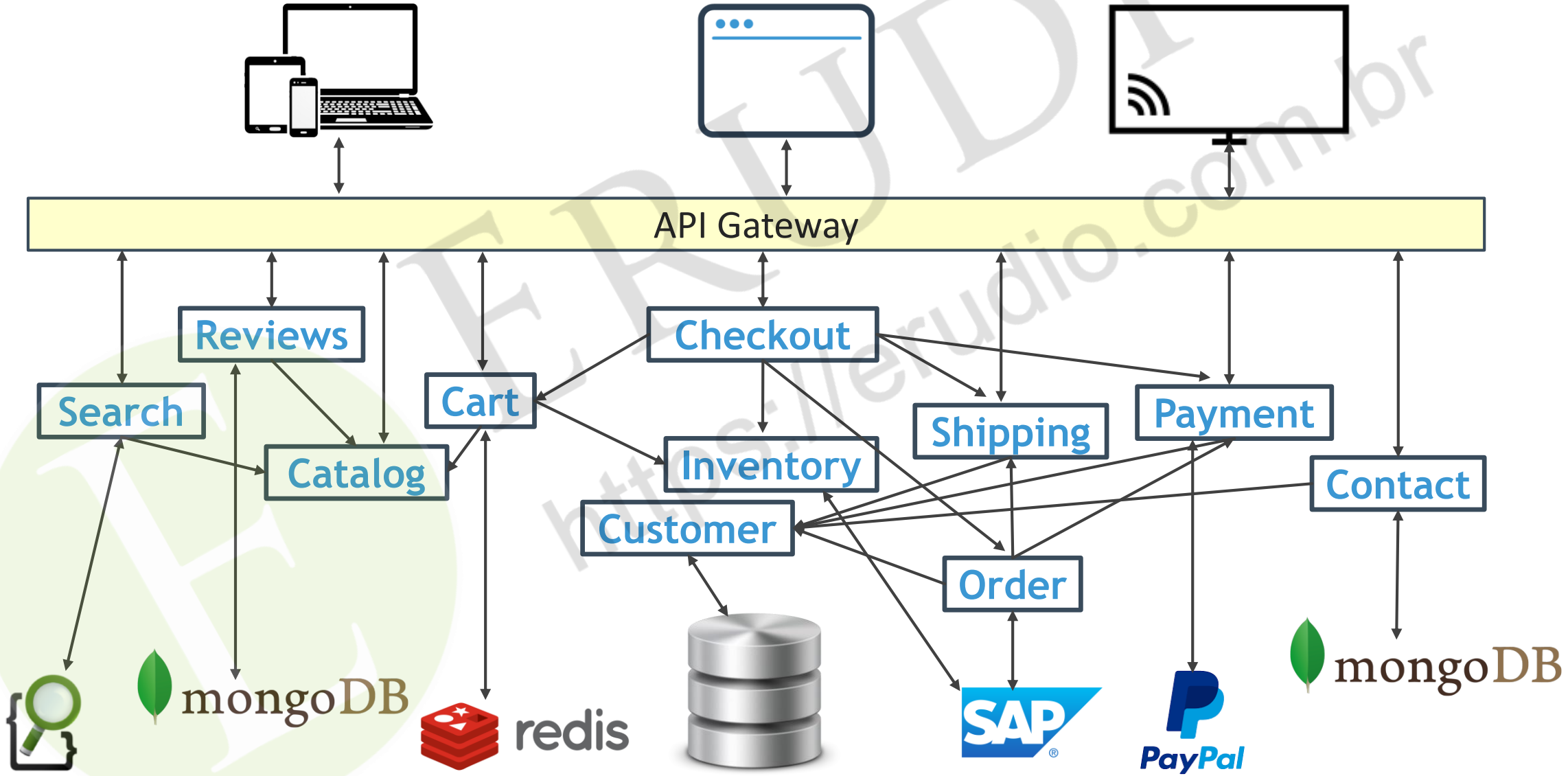
Entendendo uma Arquitetura de Microsserviços



Entendendo uma Arquitetura de Microsserviços

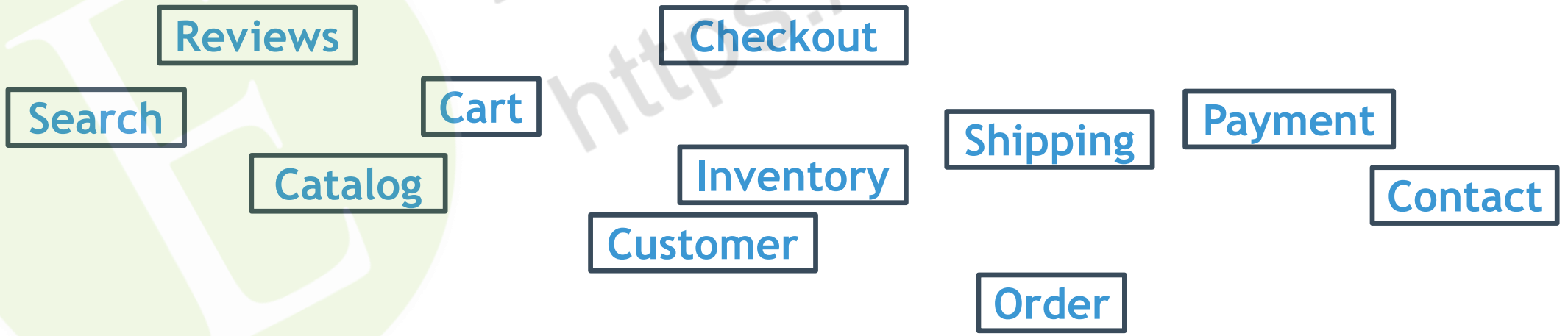


Entendendo uma Arquitetura de Microsserviços



Componentização por meio de serviços

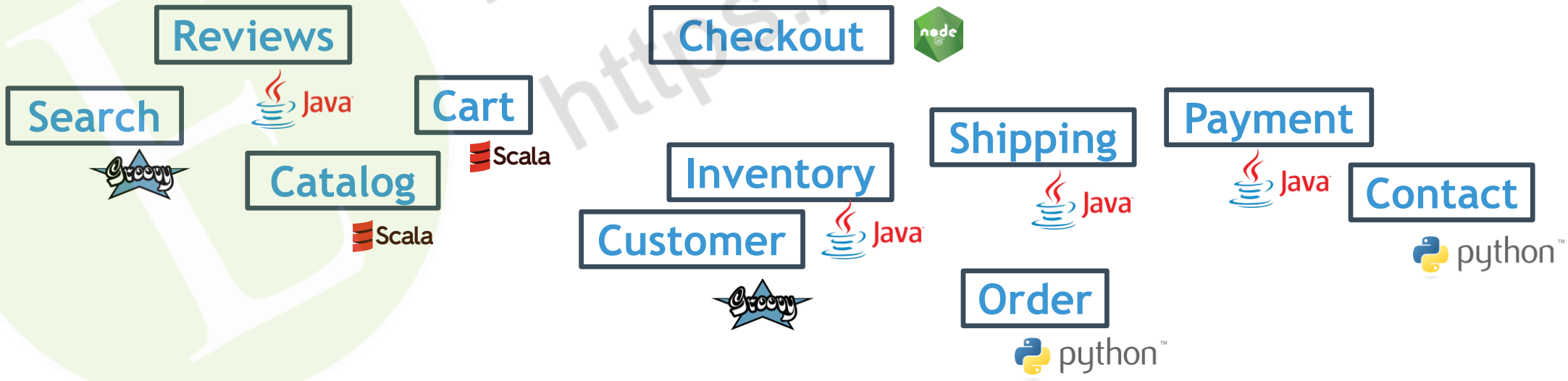
- Independente das estruturas de linguagem;
- Onde os serviços são pequenos, aplicações implantáveis de forma independente;
- Força o design de interfaces claras;
- Mudanças com escopo no serviço afetado.



Microsserviços:

(Uma combinação de diversos serviços menores)

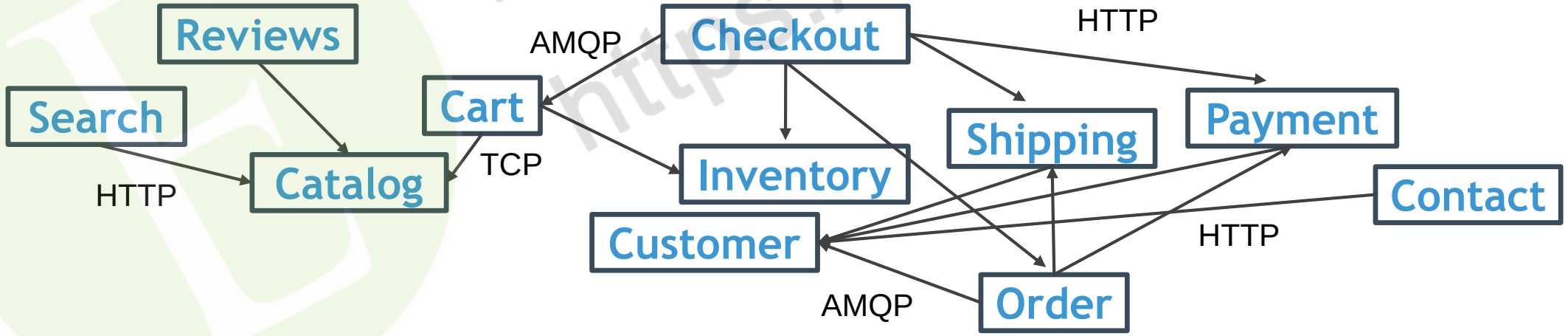
- Os serviços são pequenas aplicações implantáveis de forma independente
  - Podem não possuir uma única base de código
  - Não (necessariamente) uma única linguagem / framework



Microsserviços:

(Comunicação baseada em protocolos leves)

- HTTP, TCP, UDP, Menssageria, etc;
  - Payloads: JSON, BSON, XML, Protocol Buffers, etc
- Força o design de interfaces claras
- Arquitetura nativa para a nuvem da Netflix - comunicando-se por meio de API's
  - Não tem um banco de dados comum



Microsserviços:

(Os serviços englobam as necessidades de negócios)

- Não é baseado em uma stack de tecnologias
- Fatias verticais de acordo com as funcionalidades ou regras de negócio (ou seja, carrinho, catálogo, checkout)
- ... Ainda podemos ter serviços que realizam tarefas puramente técnicas (serviço de e-mail)
- Adequado para equipes multifuncionais

Search

PUT /search

Reviews

GET /review/123
POST /review

Cart

POST /cart
GET /cart/123
POST /cart/123/item
DELETE /cart/123
PUT /cart/123/item/1
DELETE /cart/123/item/1

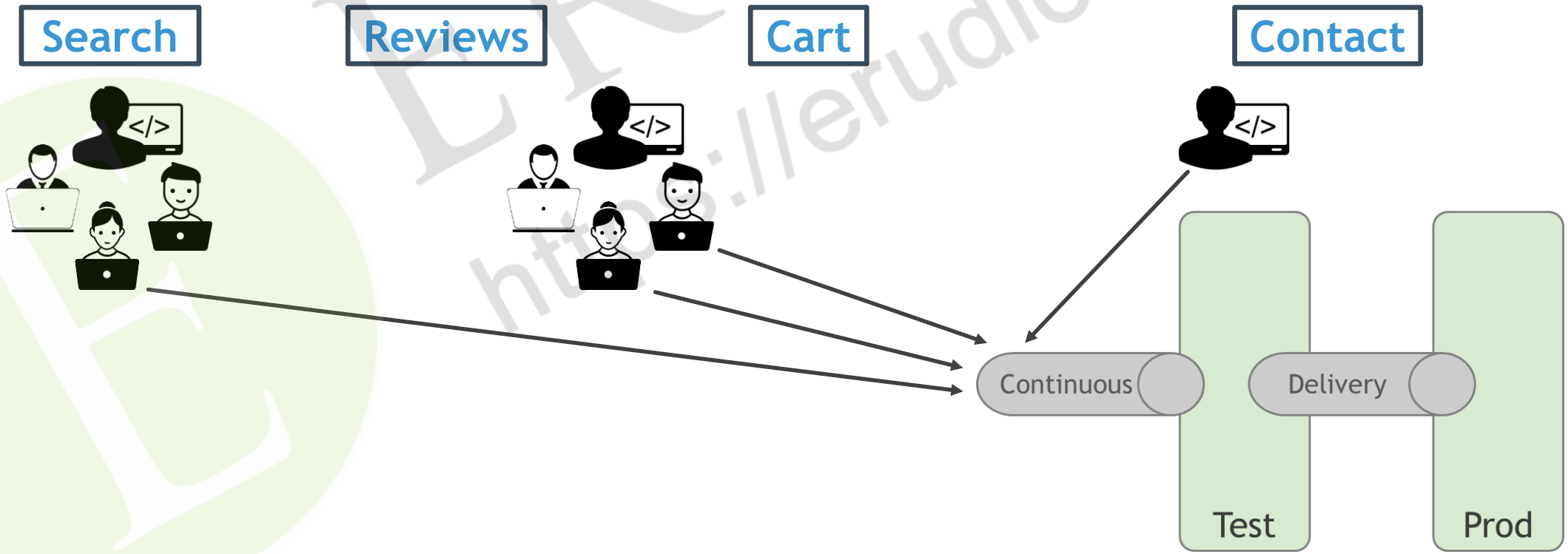
Contact

GET /post/123
POST /post

Microsserviços:

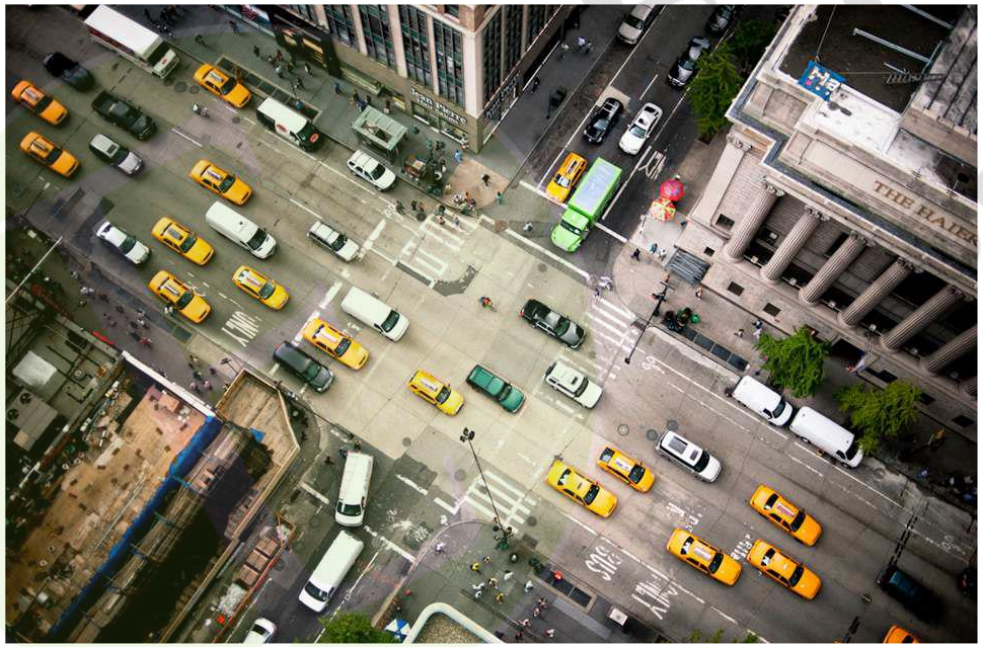
(Serviços facilmente gerenciaveis)

- Fáceis de compreender, alterar, testar, versionar, implantar, gerenciar, refatorar ou mesmo substituir
  - Por equipes pequenas e multifuncionais (ou mesmo indivíduos)



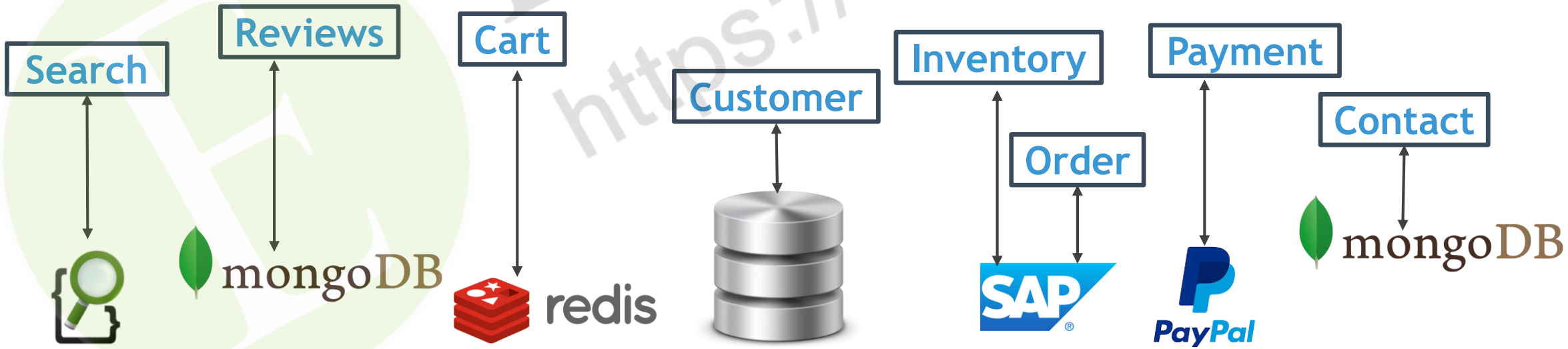
Governança Descentralizada

- Usa a ferramenta mais adequada (linguagem, framework) para cada serviço;
- Os serviços evoluem em velocidades diferentes, são implantados e gerenciados de acordo com as diferentes necessidades.
- Faça com que os serviços sejam "Tolerant Readers"
- Contratos Consumer-Driven
- Antítese da ESB
  - Os serviços não são *orquestrados*, mas sim *coreografados*



Persistência Poliglota

- Liberdade para usar a ferramenta mais adequada para cada serviço;
  - Os bancos de dados relacionais nem sempre são a melhor escolha
  - Pode ser muito controverso! Muitos DBA's estão não irão gostar!
    - Ter todos os dados concentrados em um único banco de dados relacional;
    - Não existem transações!



Vantagens dos Microsserviços

- Fácil de compreender (difícil de compreender o todo);
- Muito fácil de testar, implantar, gerenciar, versionar e escalar serviços únicos;
- Ciclo de mudanças desacoplado;
- Mais fácil de escalar os times de desenvolvimento;
- Liberdade para experimentar novas linguagens ou frameworks.

Desafios dos Microsserviços

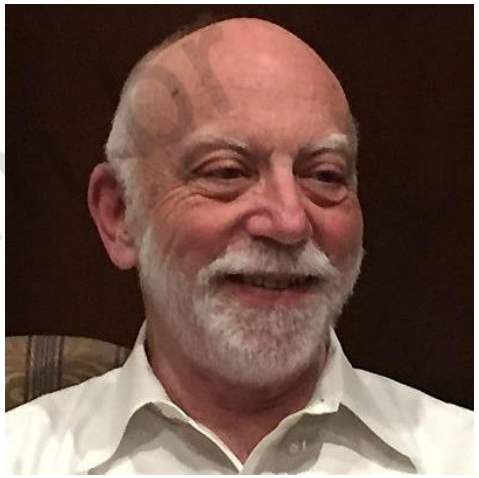
- A complexidade foi movida para fora da aplicação e foi distribuída entre os microsserviços;
  - Falácias da computação distribuída;
- Os serviços podem ficar eventualmente indisponíveis;
  - Nunca precisamos nos preocupar com isso em aplicações monolíticas;
  - Projetado para falhas, circuit breakers;
    - *"Tudo falha o tempo todo"* - Werner Vogels, CTO da Amazon.
  - É necessário muito mais monitoramento;
- Chamadas remotas mais caras do que chamadas de métodos;

Desafios dos Microsserviços

- Transações: devem depender de consistência eventual adeus ACID;
- As funcionalidades estão distribuídas por vários serviços;
- A gestão da mudança torna-se um desafio diferente;
  - Precisa considerar a interação entre os serviços;
  - Gerenciamento de dependências/versões.
- O refactory irá depender dos Module Boundaries.

Falácias da Computação Distribuída

1. A rede é confiável
2. Latência é zero
3. Largura de banda infinita
4. A rede é segura
5. A Topologia não muda
6. Existe um administrador
7. Custo de transporte é zero
8. A rede é homogênea



Laurence Peter Deutsch
Sun Microsystems

Como Quebrar uma Aplicação Monolítica em Microsserviços

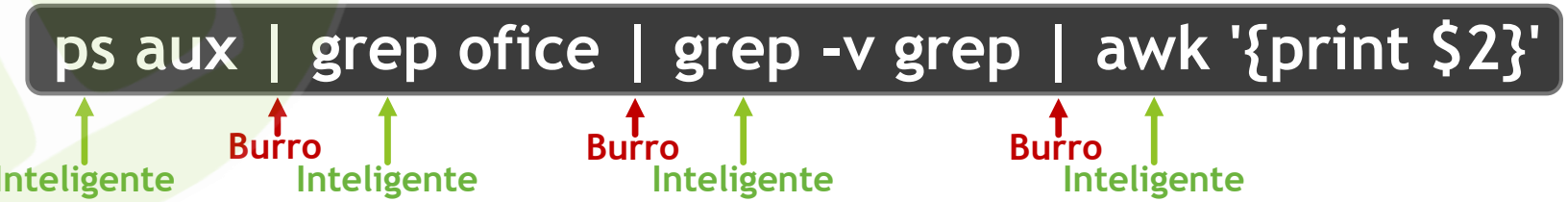
- Principal ponto a se considerar deve ser a funcionalidade de negócio
  - De acordo com o substantivo (catalog, cart, customer)
  - De acordo com o verbo (search, checkout, shipping)
  - Princípio da responsabilidade única (Single Responsibility Principle)
    - https://blog.cleancoder.com/uncle-bob/2014/05/08/SingleReponsibilityPrinciple.html
  - Princípio do contexto limitado (Bounded Context)
    - https://martinfowler.com/bliki/BoundedContext.html

O quão Micro é Micro?

- O tamanho não é o fator determinante;
  - Pequeno o suficiente para um desenvolvedor individual assimilar;
  - Pequeno o suficiente para ser construído e gerenciado por um time pequeno;
    - Regra das duas pizzas da Amazon.
  - Documentação pequena o suficiente para ler e entender;
    - Razoável → *Estatuto do Idoso* edição de 218 - 56 páginas;
    - Exagerado → *Constituição Federal* edição de 2021 - 328 páginas.
  - Dezenas de segredos, não centenas;
  - Previsível. Fácil de experimentar.

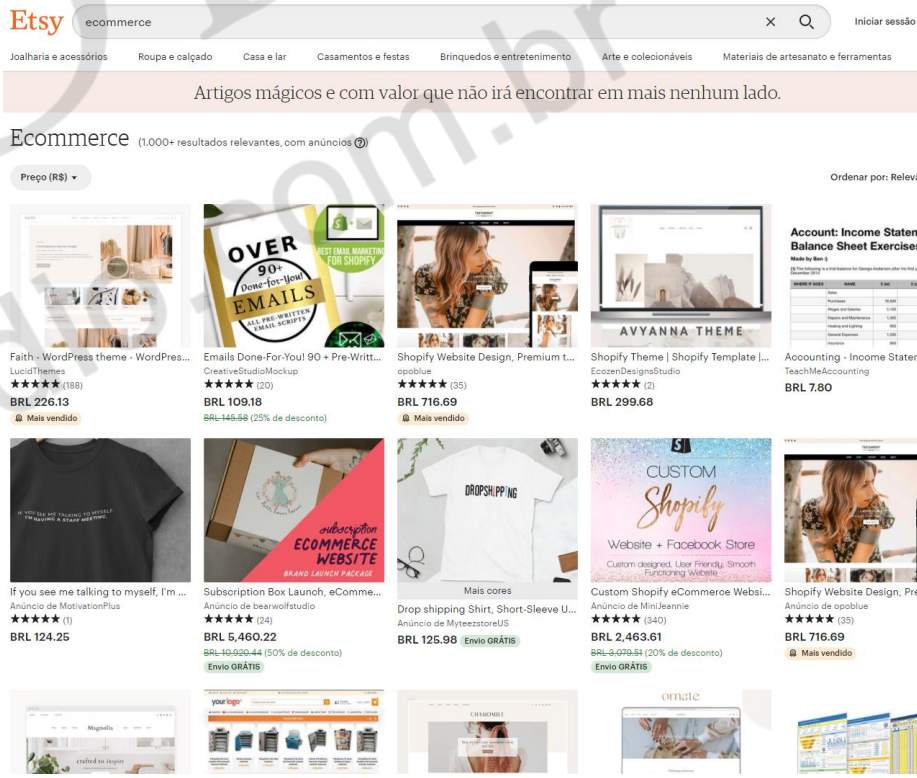
Diferenças entre Microsserviços e SOA

- SOA trata da integração entre sistemas;
  - Os microsserviços atendem a aplicativos individuais;
- SOA depende de orquestração;
  - Os microsserviços contam com coreografia;
- SOA depende de tecnologia de integração inteligente, serviços burros
  - Os microsserviços contam com serviços inteligentes e tecnologia de integração burra;
  - Considere: comandos, pipes and filters



Os Monólitos Sempre são ruins?

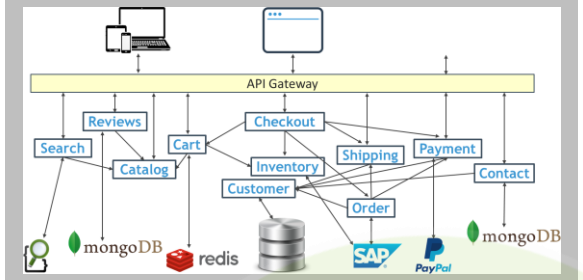
- O exemplo do e-commerce https://www.etsy.com/market/ecommerce;
  - Em 2018 a Etsy teve bilhões de visualizações de páginas, milhões de itens vendidos e milhões de membros;
  - 600 desenvolvedores implantam um único WAR 60 vezes por dia;
  - Práticas:
    - CI; Push button deployment; Bom monitoramento; Os desenvolvedores implantam no site no primeiro dia; VMs por desenvolvedor; GitHub; Chef; IRC para controlar lançamentos; dashboards; sem branches no sistema de controle de versões.



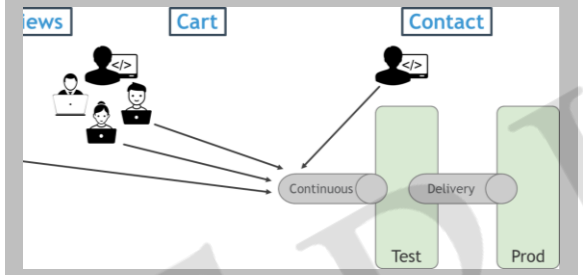
Recapitulando

"Uma aplicação única, desenvolvida como um conjunto de pequenos serviços, cada um executando o seu próprio processo e se comunicando com mecanismos leves, geralmente por API's REST através do protocolo HTTP".

– Martin Fowler



O que são Microsserviços



As Características chave dos Microsserviços

Monólitos x Microsserviços

Os Prós e Contras dos Microsserviços

Na próxima seção

...

Configurações personalizadas em Spring Boot

Spring Boot Actuator

Spring Cloud Config Server
- Local
- Github

Integrando microsserviços ao servidor de configurações